Bahrain Polytechnic بوليتكنك البحرين

Assessment Cover Sheet

Assessment Title	Database 1 (group project)		
Assessment Type	Controlled	Individual	Must-pass
Due Date	23-May-2024	**Course Code**	IT6005
Course Title	Database Systems 1		
Internal Moderator's Name	Faustino Reyes		
External Examiner's Name			

Instructions:

1. This cover sheet must be completed (section in red below) and attached to your assessment before submission in hard copy/soft copy.
2. The time allowed for this assessment is 85 days.
3. This assessment carries 155 marks distributed to a total of 4 questions assessing CILO 1,2,3,4,5.
4. The materials allowed for use in this assessment are course material and research material.
5. The **use of generative AI tools is strictly prohibited**.
6. References consulted (if any) must be properly acknowledged and cited.
7. The assessment has a total of 8 pages.

Learner ID	202302237-202304464-	**Date Submitted**	25/5/2024
Learner Name	Affan Aslam-Zekariya-Ali Abbas – Muhammed Ali – Saad Khurram		
Programme Code	IT6005		
Programme Title	DATABASE SYSTEMS 1		
Lecturer's Name	Dr Fatima Baqer		

By submitting this assessment for marking, I affirm that this assessment is my own work.

Learner Signature	Affan!

Do not write beyond this line. For assessor use only.

Assessor's Name			
Marking Date		**Maks Obtained**	

Comments:

Contents

..... **Error! Bookmark not defined.**

Database Systems Project Overview 3

Deliverables 4

Case Study: Database System 5

Task 1 and 2 ERD Modelling 7

Task 3 Research Scenario (400 to 450 words): 7

Task 4 SQL Statements 8

Draft

Database Systems Project Overview

The project is divided into three tasks:

- Task 1: Context ERD
- Task 2: Physical ERD
- Task 3: Research Question
- Task 4: SQL Statements

Due Dates:

- **Thursday 23-May-2024 at 11:55 p.m**.

Assessment Type:

- Group Project (5 members per group maximum)
- Achievement based

Weighting:

Section	Marks	LO
Task 1: Context ERD	41	1, 2
Task 2: Physical ERD	38	1, 2, 3
Task 3: Research Question	9	5
Task 4: SQL Statements	67	3,4
Total	**155**	

Late Rule:

The maximum grade granted for late submission is 60% for up to 3 calendar days. A grade of 0 will be allocated for submissions after 3 days.

Draft

Deliverables

Report

The report should include 4 sections:

- **Section 1**: a screenshot of the Context ERD (clear Diagram)
- **Section 2**:
    - screenshot of the Physical ERD (clear Diagram)
    - database implementation code
    - a description of Many to Many relationship resolutions
- **Section 3**: Research Question
- **Section 4**:
    - Insert statements to add sample rows
    - Clearly numbered SQL statements
    - Screenshot of the output from each of the SQL statements.
- Report name: **studentID1_studentID2_studentID3_studentID4_report.pdf**
- Report format: PDF

Models Folder

- This folder should include the context ERD and Physical ERD Power Architect files.
- Individual SQL files labeled as statement1.sql, statment2.sql, etc...
- Folder name: **studentID1_studentID2_studentID3_studentID4_models**

Submission

- When submitting the deliverables of the project, you must upload ***only one ZIP file***, containing the report and Models Folder.
- Note, only (.zip) compressed files will be accepted. (.rar) files will not be accepted
- Zipped file name: "**studentID1_studentID2_studentID3_studentID4_dbProject**"
- For example: "2020XXXXX_2020XXXXX _2020XXXXX _2020XXXXX _dbProject.zip"

Draft

Case Study: Database System – CityWheelsBH App

In a lively city like those in Bahrain, there's a popular app called "CityWheelsBH" that's making it easier for people to travel around. This app is a bridge between people who need a ride and those who can offer one.

Let's explore a typical day with CityWheelsBH through the experiences of some local Bahrainis:

Ali has recently begun his journey as a CityWheelsBH driver. He carefully entered his details into the app, including his name, contact number, and preferred method for receiving payments. Ali also had to show his driving license and insurance documents, important for proving he's a safe and legal driver. After each day of driving, Ali eagerly checks his rating in the app. This rating, given by the passengers he drives, tells him how well he is doing and helps him get more rides. Depending on the availability of vehicles, CityWheelsBH assigns Ali to any of the available vehicles in its inventory. Management at CityWheelsBH also keeps detailed records which of each vehicle's maintenance to prevent any unforeseen event. CityWheelsBH also keeps records of each of its drivers schedule for efficient ride assignment and vehicle utilization.

On the other side of the app is Fatima, who regularly uses CityWheelsBH for her daily trips. She's known for being a respectful and punctual passenger, which is reflected in her high passenger rating. Each morning, Fatima opens the app, selects her location and where she wants to go, often choosing a comfortable car type like a sedan for her ride. The system then automatically assigns drivers based on availability, proximity, and preferences using its database, updating both parties with the ride details for confirmation or cancellation. When Fatima reaches her destination, the payment for the ride is automatically made through her stored credit card details in the app. It's convenient and quick. Later, she leaves a rating and some feedback about her ride experience, which helps Ali and CityWheelsBH know about their service quality. In the evenings, Fatima sometimes browses through her past rides in the app's history section. She fondly remembers various journeys she's made, like a special trip to a friend's wedding, all facilitated by CityWheelsBH.

Hassan, another user of the app, recently had a problem with a ride. To resolve this, he used CityWheelsBH to submit a support request, describing his issue in hopes of getting it fixed. This feature is crucial in maintaining a trustworthy and reliable service for everyone.

CityWheelsBH also knows how to keep its users happy. They often release special promotions and discounts. For example, Mariam, who just started using the app, was delighted to find a promo code for a discount on her first ride.

Then there's the Al Mashaq family, who planned a family outing to a popular park. They needed more space and a child seat for their youngest member, so they chose to book a larger vehicle through CityWheelsBH. The pricing for their ride was dynamically calculated, considering factors like the larger vehicle type and the additional requirement of a child seat. This approach ensured that the Al Masherq family was charged a fair and accurate rate, tailored to their specific needs, enhancing their overall experience with both comfort and cost-effectiveness.

At the end of each day, the CityWheelsBH team diligently examines all the ride-specific feedback and ratings, as well as the general user feedback gathered throughout the day. This includes insights from the day's journeys, addressing both the performance of individual rides and the overall user experience with the app. They utilize this comprehensive feedback to enhance the service continually, ensuring it remains attuned to the evolving needs and expectations of its users.

In this vibrant city, each trip, each user's feedback, and every interaction on CityWheelsBH contributes to a larger picture. The app brings together drivers, passengers, their journeys, the payments, and the feedback in a well-

Draft

organized system. This scenario showcases how everything works together to provide a reliable and convenient service for people in Bahrain.

Problem Statement:

You will need to design a database to be used for the back end of the above-mentioned mobile app. CityWheelsBH is a new startup company and do not have any infrastructure in place yet. When you design your model take into consideration the growth of the company and produce a model that is normalized.

If you discover that the data in the case study is incomplete, you are expected to add necessary assumptions. You are also expected to include suggestions that would improve the system and justify your assumptions. Decide on relevant attributes for your tables.

Draft

Task 1 and 2 ERD Modelling

In these tasks, you must create a Context ERD (Task 1) and Physical ERD (Task 2) for CityWheelsBH. The designs should be presented in a report and the respective Power Architect files should also be attached.

Task 1

- Design Context ERD
- Identify at least **10 entities**
- Identify all the **relationships** between the entities. This should include at least **one M:N relationship**
- Identify the correct **cardinalities** for each entity

Task 2

- Design Physical ERD
- Include relevant key and non-key attributes for each entity
- Include data type and size for each attribute
- Resolve the M:N relationships and add a description of them in the report (at least one)
- **Generate database implementation script**. This should include code to implement at least 10 entities along with relevant attributes and key constraints.

Task 3 Research Scenario (400 to 450 words):

Consider the scenario where CityWheelsBH, after experiencing substantial growth, plans to expand its operations to other GCC countries and eventually globally. As part of this expansion, the company contemplates sharing the extensive personal data it has accumulated about its passengers with its new international branches. Analyze the legal and ethical implications of CityWheelsBH sharing this data across borders.

In your discussion, reference specific data protection laws both in Bahrain and from an international perspective. For instance, how would Bahrain's Personal Data Protection Law (PDPL) influence such a decision? Compare this with international regulations like the General Data Protection Regulation (GDPR) or The Uniform Personal Data Protection Act.

Examine the nuances of legal compliance, including the necessity for consent and the ethical considerations of protecting passenger privacy. What are the responsibilities of CityWheelsBH in ensuring data security and privacy in this context?

Support your analysis with at least two references, adhering to the APA referencing format. Make sure to include at least one law from Bahrain and one international law in your discussion.

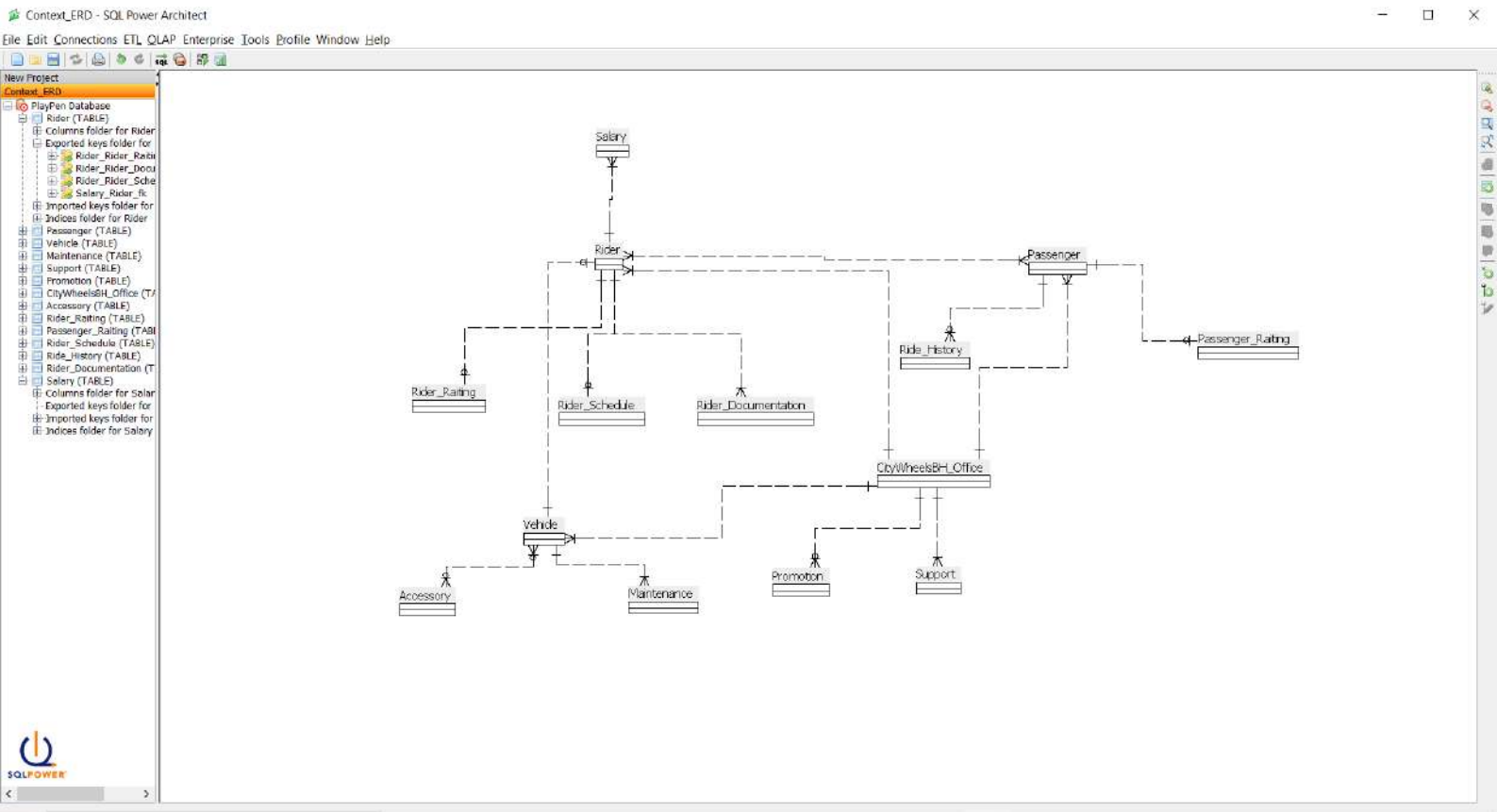
Draft

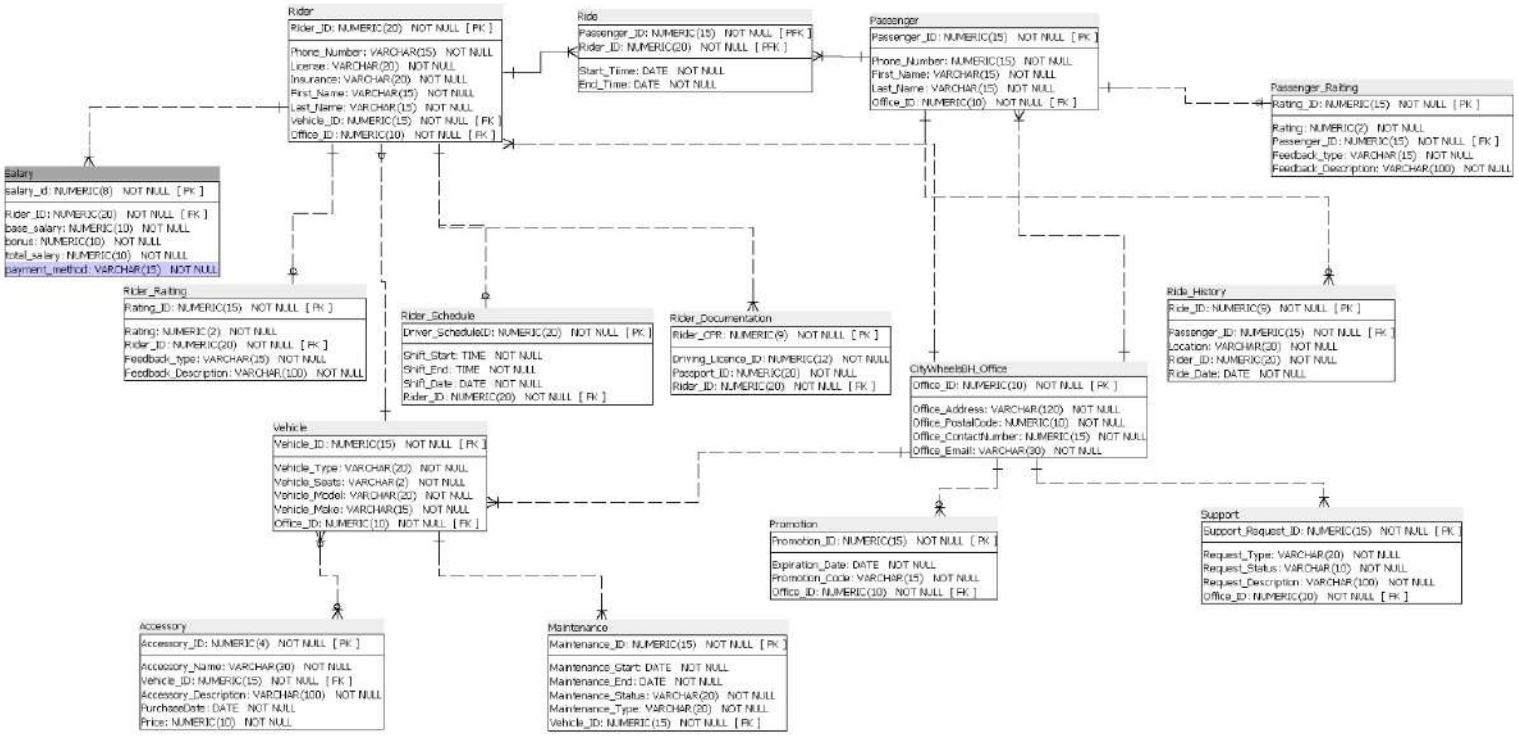
Task 4 SQL Statements

In this section you will need to take the code you generated from your design and run it on SQL developer to create all of your tables. You will then need to populate all of these tables with sample data. To each table you will need to add at least 5 sample rows of data. After doing so, you will need to write a single SQL statement for each of the given tasks below. In your report you will need to include all of your insert statements and all of the SQL statements you have written numbered correctly. You will also provide screenshots of the output from each query.

1. List drivers and the total number of rides they have completed.
2. Find the average rating of drivers who have completed more than 50 rides.
3. List the names of passengers who have used CityWheelsBH more than 20 times.
4. Identify drivers who have never received a rating below 4.
5. Find the total number of rides completed in each month of the current year.
6. Show drivers who have completed rides with at least three different passengers.
7. Determine the most popular destination among all rides.
8. Identify the most common issue reported in support tickets.
9. Show the total earnings for each driver in the current year.
10. Get a list of all promotions that are currently active and expire within the next 30 days.

Draft


Context_ERD - SQL Power Architect
File Edit Connections ETL OLAP Enterprise Tools Profile Window Help
New Project
Context_ERD
PlayPen Database
Rider (TABLE)
Columns folder for Rider
Exported keys folder for
Imported keys folder for
Indices folder for Rider
Passenger (TABLE)
Vehicle (TABLE)
Maintenance (TABLE)
Support (TABLE)
Promotion (TABLE)
Accessory (TABLE)
Rider_Raiting (TABLE)
Ride_History (TABLE)
Salary (TABLE)
Columns folder for Salar
Exported keys folder for
Imported keys folder for
Indices folder for Salary
SQLPOWER
Salary
Rider
Passenger
Ride_History
Passenger_Raiting
Rider_Raiting
Rider_Schedule
Rider_Documentation
CityWheelsBH_Office
Vehicle
Promotion
Support
Accessory
Maintenance


Rider
Rider_ID: NUMERIC(20) NOT NULL [PK]
Phone_Number: VARCHAR(15) NOT NULL
License: VARCHAR(20) NOT NULL
Insurance: VARCHAR(20) NOT NULL
First_Name: VARCHAR(15) NOT NULL
Last_Name: VARCHAR(15) NOT NULL
Vehicle_ID: NUMERIC(15) NOT NULL [FK]
Office_ID: NUMERIC(10) NOT NULL [FK]
Ride
Passenger_ID: NUMERIC(15) NOT NULL [PFK]
Rider_ID: NUMERIC(20) NOT NULL [PFK]
Start_Time: DATE NOT NULL
End_Time: DATE NOT NULL
Passenger
Passenger_ID: NUMERIC(15) NOT NULL [PK]
Phone_Number: NUMERIC(15) NOT NULL
First_Name: VARCHAR(15) NOT NULL
Last_Name: VARCHAR(15) NOT NULL
Office_ID: NUMERIC(10) NOT NULL [FK]
Passenger_Rating
Rating_ID: NUMERIC(15) NOT NULL [PK]
Rating: NUMERIC(2) NOT NULL
Passenger_ID: NUMERIC(15) NOT NULL [FK]
Feedback_type: VARCHAR(15) NOT NULL
Feedback_Description: VARCHAR(100) NOT NULL
Salary
salary_id: NUMERIC(8) NOT NULL [PK]
Rider_ID: NUMERIC(20) NOT NULL [FK]
base_salary: NUMERIC(10) NOT NULL
bonus: NUMERIC(10) NOT NULL
total_salary: NUMERIC(10) NOT NULL
payment_method: VARCHAR(15) NOT NULL
Rider_Rating
Rating_ID: NUMERIC(15) NOT NULL [PK]
Rating: NUMERIC(2) NOT NULL
Rider_ID: NUMERIC(20) NOT NULL [FK]
Feedback_type: VARCHAR(15) NOT NULL
Feedback_Description: VARCHAR(100) NOT NULL
Rider_Schedule
Driver_ScheduleID: NUMERIC(20) NOT NULL [PK]
Shift_Start: TIME NOT NULL
Shift_End: TIME NOT NULL
Shift_Date: DATE NOT NULL
Rider_ID: NUMERIC(20) NOT NULL [FK]
Rider_Documentation
Rider_CPR: NUMERIC(9) NOT NULL [PK]
Driving_Licence_ID: NUMERIC(12) NOT NULL
Passport_ID: NUMERIC(20) NOT NULL
Rider_ID: NUMERIC(20) NOT NULL [FK]
Ride_History
Ride_ID: NUMERIC(9) NOT NULL [PK]
Passenger_ID: NUMERIC(15) NOT NULL [FK]
Location: VARCHAR(30) NOT NULL
Rider_ID: NUMERIC(20) NOT NULL
Ride_Date: DATE NOT NULL
CityWheelsGH_Office
Office_ID: NUMERIC(10) NOT NULL [PK]
Office_Address: VARCHAR(120) NOT NULL
Office_PostalCode: NUMERIC(10) NOT NULL
Office_ContactNumber: NUMERIC(15) NOT NULL
Office_Email: VARCHAR(30) NOT NULL
Vehicle
Vehicle_ID: NUMERIC(15) NOT NULL [PK]
Vehicle_Type: VARCHAR(20) NOT NULL
Vehicle_Seats: VARCHAR(2) NOT NULL
Vehicle_Model: VARCHAR(20) NOT NULL
Vehicle_Make: VARCHAR(15) NOT NULL
Office_ID: NUMERIC(10) NOT NULL [FK]
Promotion
Promotion_ID: NUMERIC(15) NOT NULL [PK]
Expiration_Date: DATE NOT NULL
Promotion_Code: VARCHAR(15) NOT NULL
Office_ID: NUMERIC(10) NOT NULL [FK]
Support
Support_Request_ID: NUMERIC(15) NOT NULL [PK]
Request_Type: VARCHAR(20) NOT NULL
Request_Status: VARCHAR(10) NOT NULL
Request_Description: VARCHAR(100) NOT NULL
Office_ID: NUMERIC(10) NOT NULL [FK]
Accessory
Accessory_ID: NUMERIC(4) NOT NULL [PK]
Accessory_Name: VARCHAR(30) NOT NULL
Vehicle_ID: NUMERIC(15) NOT NULL [FK]
Accessory_Description: VARCHAR(100) NOT NULL
PurchaseDate: DATE NOT NULL
Price: NUMERIC(10) NOT NULL
Maintenance
Maintenance_ID: NUMERIC(15) NOT NULL [PK]
Maintenance_Start: DATE NOT NULL
Maintenance_End: DATE NOT NULL
Maintenance_Status: VARCHAR(20) NOT NULL
Maintenance_Type: VARCHAR(20) NOT NULL
Vehicle_ID: NUMERIC(15) NOT NULL [FK]

An M:N (many-to-many) relationship in database enables the connection of multiple columns from one table to multiple columns from another table. This type of relationship is important for modeling complex, real-world interactions between entities.

As in this ERD we had Rider and Passenger M:N relationship. To solve it we made another table to connect a rider with his passenger. With its help we can find the details of the ride the passenger did with that rider and all details about it. By using rider and passenger PKs

```
CREATE TABLE CityWheelsBH_Office (
        Office_ID NUMBER(10) NOT NULL,
        Office_Address VARCHAR2(120) NOT NULL,
        Office_PostalCode NUMBER(10) NOT NULL,
        Office_ContactNumber NUMBER(15) NOT NULL,
        Office_Email VARCHAR2(30) NOT NULL,
        CONSTRAINT CITYWHEELSBH_OFFICE_ID_PK PRIMARY KEY (Office_ID)
);


CREATE TABLE Promotion (
        Promotion_ID NUMBER(15) NOT NULL,
        Expiration_Date DATE NOT NULL,
        Promotion_Code VARCHAR2(15) NOT NULL,
        Office_ID NUMBER(10) NOT NULL,
        CONSTRAINT PROMOTION_ID_PK PRIMARY KEY (Promotion_ID)
);


CREATE TABLE Support (
        Support_Request_ID NUMBER(15) NOT NULL,
        Request_Type VARCHAR2(20) NOT NULL,
        Request_Status VARCHAR2(10) NOT NULL,
        Request_Description VARCHAR2(100) NOT NULL,
        Office_ID NUMBER(10) NOT NULL,
        CONSTRAINT SUPPORT_ID_PK PRIMARY KEY (Support_Request_ID)
);
```

```
CREATE TABLE Vehicle (
        Vehicle_ID NUMBER(15) NOT NULL,
        Vehicle_Type VARCHAR2(20) NOT NULL,
        Vehicle_Seats VARCHAR2(2) NOT NULL,
        Vehicle_Model VARCHAR2(20) NOT NULL,
        Vehicle_Make VARCHAR2(15) NOT NULL,
        Office_ID NUMBER(10) NOT NULL,
        CONSTRAINT VEHICLE_ID_PK PRIMARY KEY (Vehicle_ID)
);


CREATE TABLE Maintenance (
        Maintenance_ID NUMBER(15) NOT NULL,
        Maintenance_Start DATE NOT NULL,
        Maintenance_End DATE NOT NULL,
        Maintenance_Status VARCHAR2(20) NOT NULL,
        Maintenance_Type VARCHAR2(20) NOT NULL,
        Vehicle_ID NUMBER(15) NOT NULL,
        CONSTRAINT MAINTENANCE_ID_PK PRIMARY KEY (Maintenance_ID)
);


CREATE TABLE Accessory (
        Accessory_ID NUMBER(4) NOT NULL,
        Accessory_Name VARCHAR2(30) NOT NULL,
        Vehicle_ID NUMBER(15) NOT NULL,
        Accessory_Description VARCHAR2(100) NOT NULL,
        PurchaseDate DATE NOT NULL,
        Price NUMBER(10) NOT NULL,
```

```
        CONSTRAINT ACCESSORY_ID_PK PRIMARY KEY (Accessory_ID)
);


CREATE TABLE Passenger (
        Passenger_ID NUMBER(15) NOT NULL,
        Phone_Number NUMBER(15) NOT NULL,
        First_Name VARCHAR2(15) NOT NULL,
        Last_Name VARCHAR2(15) NOT NULL,
        Office_ID NUMBER(10) NOT NULL,
        CONSTRAINT PASSENGER_ID_PK PRIMARY KEY (Passenger_ID)
);


CREATE TABLE Ride_History (
        Ride_ID NUMBER(9) NOT NULL,
        Passenger_ID NUMBER(15) NOT NULL,
        Location VARCHAR2(30) NOT NULL,
        Rider_ID NUMBER(20) NOT NULL,
        Ride_Date DATE NOT NULL,
        CONSTRAINT RIDE_HISTORY_ID_PK PRIMARY KEY (Ride_ID)
);


CREATE TABLE Passenger_Raiting (
        Rating_ID NUMBER(15) NOT NULL,
        Rating NUMBER(2) NOT NULL,
        Passenger_ID NUMBER(15) NOT NULL,
        Feedback_type VARCHAR2(15) NOT NULL,
```

```
        Feedback_Description VARCHAR2(100) NOT NULL,
        CONSTRAINT PASSENGER_RAITING_ID_PK PRIMARY KEY (Rating_ID)
);


CREATE TABLE Rider (
        Rider_ID NUMBER(20) NOT NULL,
        Phone_Number VARCHAR2(15) NOT NULL,
        License VARCHAR2(20) NOT NULL,
        Insurance VARCHAR2(20) NOT NULL,
        First_Name VARCHAR2(15) NOT NULL,
        Last_Name VARCHAR2(15) NOT NULL,
        Vehicle_ID NUMBER(15) NOT NULL,
        Office_ID NUMBER(10) NOT NULL,
        CONSTRAINT RIDER_ID_PK PRIMARY KEY (Rider_ID)
);


CREATE TABLE Salary (
        salary_id NUMBER(8) NOT NULL,
        Rider_ID NUMBER(20) NOT NULL,
        base_salary NUMBER(10) NOT NULL,
        bonus NUMBER(10) NOT NULL,
        total_salary NUMBER(10) NOT NULL,
        payment_method VARCHAR2(15) NOT NULL,
        CONSTRAINT SALARY_PK PRIMARY KEY (salary_id)
);
```

```
CREATE TABLE Ride (
        Passenger_ID NUMBER(15) NOT NULL,
        Rider_ID NUMBER(20) NOT NULL,
        Start_Time DATE NOT NULL,
        End_Time DATE NOT NULL,
        CONSTRAINT RIDE_PK PRIMARY KEY (Passenger_ID, Rider_ID)
);


CREATE TABLE Rider_Schedule (
        Driver_ScheduleID NUMBER(20) NOT NULL,
        Shift_Start DATE NOT NULL,
        Shift_End DATE NOT NULL,
        Shift_Date DATE NOT NULL,
        Rider_ID NUMBER(20) NOT NULL,
        CONSTRAINT RIDER_SCHEDULE_ID_PK PRIMARY KEY (Driver_ScheduleID)
);


CREATE TABLE Rider_Documentation (
        Rider_CPR NUMBER(9) NOT NULL,
        Driving_Licence_ID NUMBER(12) NOT NULL,
        Passport_ID NUMBER(20) NOT NULL,
        Rider_ID NUMBER(20) NOT NULL,
        CONSTRAINT RIDER_DOCUMENTATION_ID_PK PRIMARY KEY (Rider_CPR)
);


CREATE TABLE Rider_Raiting (
```

```
    Rating_ID NUMBER(15) NOT NULL,
    Rating NUMBER(2) NOT NULL,
    Rider_ID NUMBER(20) NOT NULL,
    Feedback_type VARCHAR2(15) NOT NULL,
    Feedback_Description VARCHAR2(100) NOT NULL,
    CONSTRAINT RIDER_RAITING_ID_PK PRIMARY KEY (Rating_ID)
);


ALTER TABLE Promotion ADD CONSTRAINT CITYWHEELSBH_OFFICE_PROMOTI854
FOREIGN KEY (Office_ID)
REFERENCES CityWheelsBH_Office (Office_ID)
NOT DEFERRABLE;

ALTER TABLE Support ADD CONSTRAINT CITYWHEELSBH_OFFICE_SUPPORT_FK
FOREIGN KEY (Office_ID)
REFERENCES CityWheelsBH_Office (Office_ID)
NOT DEFERRABLE;

ALTER TABLE Passenger ADD CONSTRAINT CITYWHEELSBH_OFFICE_PASSENG671
FOREIGN KEY (Office_ID)
REFERENCES CityWheelsBH_Office (Office_ID)
NOT DEFERRABLE;

ALTER TABLE Rider ADD CONSTRAINT CITYWHEELSBH_OFFICE_RIDER_FK
FOREIGN KEY (Office_ID)
REFERENCES CityWheelsBH_Office (Office_ID)
NOT DEFERRABLE;
```

```
ALTER TABLE Vehicle ADD CONSTRAINT CITYWHEELSBH_OFFICE_VEHICLE_FK
FOREIGN KEY (Office_ID)
REFERENCES CityWheelsBH_Office (Office_ID)
NOT DEFERRABLE;

ALTER TABLE Accessory ADD CONSTRAINT VEHICLE_ACCESSORY_FK
FOREIGN KEY (Vehicle_ID)
REFERENCES Vehicle (Vehicle_ID)
NOT DEFERRABLE;

ALTER TABLE Maintenance ADD CONSTRAINT VEHICLE_MAINTENANCE_FK
FOREIGN KEY (Vehicle_ID)
REFERENCES Vehicle (Vehicle_ID)
NOT DEFERRABLE;

ALTER TABLE Rider ADD CONSTRAINT RIDER_VEHICLE_FK
FOREIGN KEY (Vehicle_ID)
REFERENCES Vehicle (Vehicle_ID)
NOT DEFERRABLE;

ALTER TABLE Passenger_Raiting ADD CONSTRAINT PASSENGER_PASSENGER_RAITING_FK
FOREIGN KEY (Passenger_ID)
REFERENCES Passenger (Passenger_ID)
NOT DEFERRABLE;

ALTER TABLE Ride_History ADD CONSTRAINT PASSENGER_RIDE_HISTORY_FK
FOREIGN KEY (Passenger_ID)
REFERENCES Passenger (Passenger_ID)
NOT DEFERRABLE;
```

```
ALTER TABLE Ride ADD CONSTRAINT PASSENGER_RIDE_FK
FOREIGN KEY (Passenger_ID)
REFERENCES Passenger (Passenger_ID)
NOT DEFERRABLE;

ALTER TABLE Rider_Raiting ADD CONSTRAINT RIDER_RIDER_RAITING_FK
FOREIGN KEY (Rider_ID)
REFERENCES Rider (Rider_ID)
NOT DEFERRABLE;

ALTER TABLE Rider_Documentation ADD CONSTRAINT RIDER_RIDER_DOCUMENTATION_FK
FOREIGN KEY (Rider_ID)
REFERENCES Rider (Rider_ID)
NOT DEFERRABLE;

ALTER TABLE Rider_Schedule ADD CONSTRAINT RIDER_RIDER_SCHEDULE_FK
FOREIGN KEY (Rider_ID)
REFERENCES Rider (Rider_ID)
NOT DEFERRABLE;

ALTER TABLE Ride ADD CONSTRAINT RIDER_RIDE_FK
FOREIGN KEY (Rider_ID)
REFERENCES Rider (Rider_ID)
NOT DEFERRABLE;

ALTER TABLE Salary ADD CONSTRAINT RIDER_SALARY_FK
FOREIGN KEY (Rider_ID)
REFERENCES Rider (Rider_ID)
```

NOT DEFERRABLE;

Saad Khurram-202200136

Task 3: The Data Protection Authority (PDA) is responsible for ensuring that personal data is processed and used according to legal standards. In March 2022, the Personal Data Protection Authority (PDPA) issued ministerial resolutions that included the rules regarding the transferring of data outside Bahrain. These rules and regulations are governed by the Ministry of Justice, Islamic Affairs, and Waqf.

The Ministry of Justice, Islamic Affairs, and Waqt lays out certain regulations that directly influence the decisions made by the company. These are:

- That the country the data is being transferred to takes all the necessary measures to protect the personal data.
- The data must be accurate, and relevant and must be kept up to date.
- The data controller shall obtain prior authorization from the Authority to transfer data outside the country.
- There should be an international agreement and legislation that data are in force in the country to which the data is transferred.

The following is the comparison between PDPL and GDRP rules and regulations:

PDPL	GDRP
1. It states that processing must be lawful, and fair and provide accurate and Relevant data	1. The GDRP also states that the nature of data must be relevant and must be kept up to date
2. The Data Controller shall obtain prior authorization from the authority to transfer data.	2. There is no need to obtain authorization from the authority.
3. The data shall not be kept longer than necessary.	3. GDRP also has storage limitations to only store necessary and relevant data.
4. Ensuring that the Data Subject can view all processed personal data, in addition to his right to rectify, block, or erase the data if the processing is not in line with the provisions of the Law.	4 The GDRP doesn't mention any of these laws directly.

Conclusion: Although, both PDPL and GDRP follow a similar set of standards. GDRP is quite strict when it comes to imposing fines on those violating it. Moreover, it has the toughest privacy and security laws which are not only limited to those in

European Union. Plus, GDRP follows a long-length structure to take measures and protect the rights and privacy of data.

1. The responsibility in ensuring data security and safety are:
   Legal compliance and consent:

- City Wheel must obtain explicit consent from the users for collecting, storing, and processing their data.
- The app should have clear terms and services that outline how data would be used, shared, and protected.

2. Ethical Considerations:
   - The data should be solely collected for the operation of the services. And should also be informed what data is being collected. City wheel should also allow the user to access the data and allow them to delete the data as per their wish.

REFERENCES

What is GDPR, the EU's new data protection law? - GDPR.eu

Bahrain Business Laws | Personal Data Protection Law

Personal Data Protection (Bahrain.bh)

INSERT STATEMENTS

```
select * from CityWheelsBH_Office;

insert into CityWheelsBH_Office values ( 1,'Street 66- Mahooz 666' , 11, 36564332 , '
office1@gmail.com' );

insert into CityWheelsBH_Office values ( 2,'Street 77- Manama 777' , 22, 97443621 , '
office2@gmail.com' );

insert into CityWheelsBH_Office values ( 3,'Street 88- Sitra 888' , 33, 39277194, '
office3@gmail.com' );

insert into CityWheelsBH_Office values ( 4,'Street 99- Tubli 999' , 44, 82837123 , '
office4@gmail.com' );

insert into CityWheelsBH_Office values ( 5,'Street 00- Isa Town 000' ,55, 08327817 , '
office5@gmail.com' );


select * from promotion;

insert into promotion values ( 21,'14-jan-2027' , 'Promo121', 1);

insert into promotion values ( 22,'13-may-2028' , 'Promo212', 2);

insert into promotion values ( 23,'20-jan-2029' , 'DayLife', 3);

insert into promotion values ( 24,'09-jul-2030' , 'SaveMoney', 4);

insert into promotion values ( 25,'25-feb-2027' ,'Promo342', 5);


select * from support;

insert into support values ( 51,'Order' , 'Active', 'Good' , 1);

insert into support values ( 52,'Maintenance' , 'Closed', 'Takes too long' , 2);

insert into support values ( 53,'Feedback' , 'Active', 'Excellent' , 3);

insert into support values ( 54,'Permission' , 'Active', 'Good' , 4);

insert into support values ( 55,'Delivery' ,'Closed', 'Fast' , 5);


select * from vehicle;

insert into vehicle values ( 2353,'Sports car' , '2', 'Dodge challenger' , 'Canada' , 1);

insert into vehicle values ( 8356,'Sports car' , '4', 'Mercedes-Benz' , 'Germany' , 2);

insert into vehicle values ( 8926,'Normal car' , '4', 'Honda crv' , 'United States' , 3);

insert into vehicle values ( 9186,'Economical car' , '4', 'Toyota corolla' , 'Japan' , 4);

insert into vehicle values ( 9345,'Bus' ,'28', 'Nissan' , 'Japan' , 5);
```

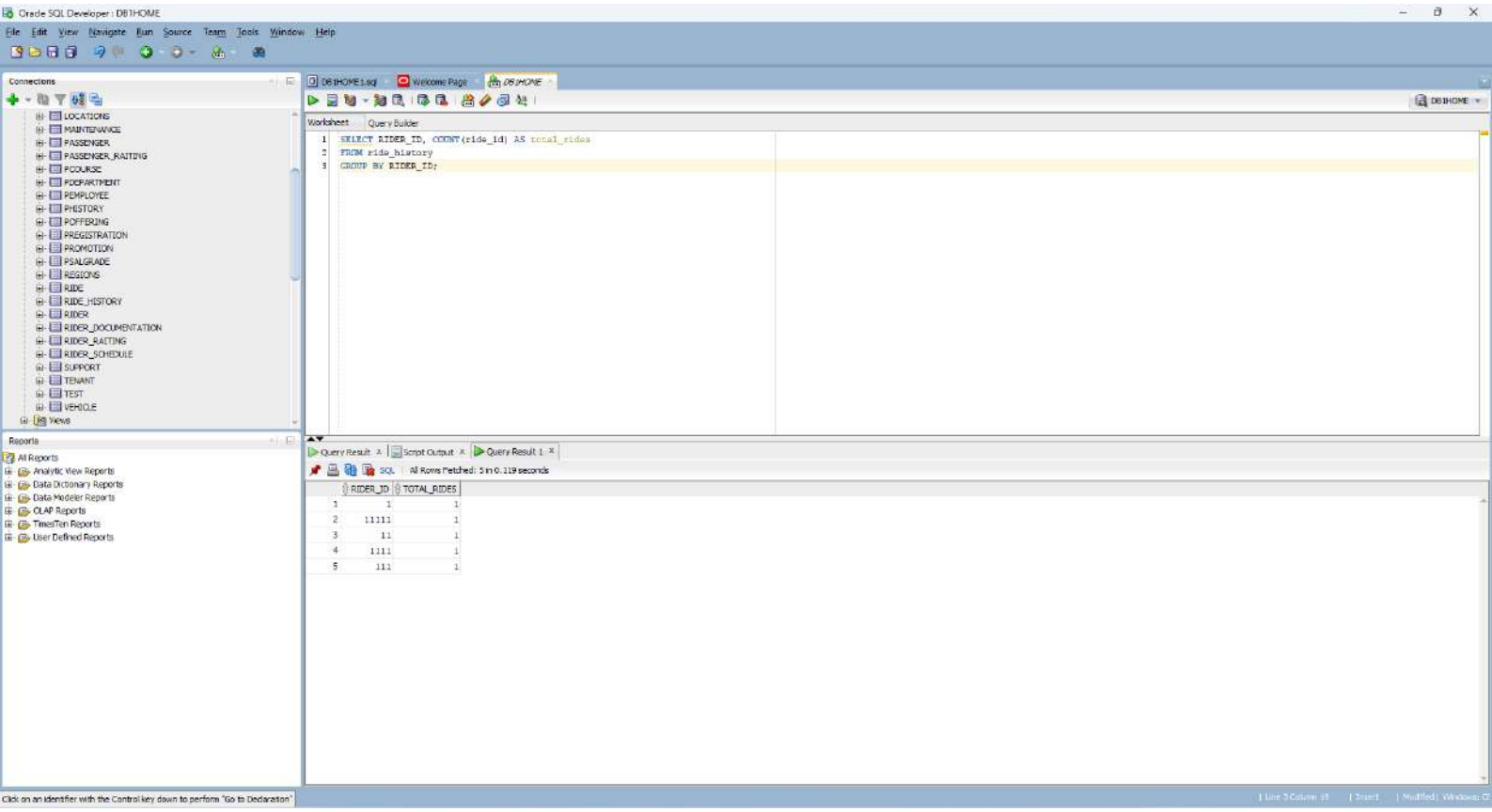
```
select * from Maintenance;

insert into Maintenance values ( 667,'22-may-2023' , '23-may-2023', 'Changing air filter' , 'Air filter' , 2353);

insert into Maintenance values ( 668,'23-may-2023' , '24-may-2023', 'Battery testing' , 'Battery' , 8356);

insert into Maintenance values ( 669,'24-may-2023' , '25-may-2023', 'Replacing belts' , 'Belts' , 8926);

insert into Maintenance values ( 700,'25-may-2023' , '26-may-2023', 'Checking oil' , 'Engine oil' , 9186);

insert into Maintenance values ( 701,'26-may-2023' , '27-may-2023', 'Replacing tires' , 'Tires' , 9345);


select * from Accessory;

insert into Accessory values ( 22,'Seat covers' , '2353', 'Plastic seat covers' , '25-feb-2023' , 11);

insert into Accessory values ( 23,'Dash camera' , '8356', 'Car camera' , '20-may-2022' , 23);

insert into Accessory values ( 24,'Seat covers' , '8926', 'Plastic seat covers' , '05-jan-2020' , 11);

insert into Accessory values ( 25,'Seat covers' , '9186', 'Plastic seat covers' , '11-may-2021' , 11);

insert into Accessory values ( 26,'Gps' , '9345', 'Gps device' , '09-dec-2019' , 13);


select * from passenger;

insert into passenger values ( 10,'111111111' , 'Ali', 'Hassan' , 1);

insert into passenger values ( 11,'222222222' , 'Mohammed', 'Hussain', 2);

insert into passenger values ( 12,'333333333' , 'Affan', 'Khan' , 3);

insert into passenger values ( 13,'444444444' , 'Saad', 'Doherty' , 4);

insert into passenger values ( 14,'555555555' , 'Ahmed', 'Abbas' , 5);


select * from passenger_raiting;

insert into passenger_raiting values ( 10,'9' , '10', 'Comment' , 'positive');

insert into passenger_raiting values ( 11,'10' , '11', 'Comment', 'positive');

insert into passenger_raiting values ( 12,'8' , '12', 'Comment' , 'normal');

insert into passenger_raiting values ( 13,'8' , '13', 'Comment' , 'positive');
```

```
insert into passenger_raiting values ( 14,'9' , '14', 'Comment' , 'positive');

select * from rider;
insert into rider values ( 1,'12345' , 'car', 'illness rider' , 'James' , 'Harron' , 2353 , 1);
insert into rider values ( 11,'123456' , 'car', 'Income rider', 'Mike' , 'Arteta' , 8356 , 2);
insert into rider values ( 111,'1234567' , 'car', 'illness rider' , 'Mikel' , 'Jhonson' , 8926 , 3);
insert into rider values ( 1111,'12345678' , 'car', 'Accidental death' , 'Jhon' , 'Harvey' , 9186 , 4);
insert into rider values ( 11111,'123456789' , 'bus', 'Family Rider' , 'Petter' , 'Potter' , 9345 , 5);

select * from ride_history;
insert into ride_history values ( 1,'10' , 'park', '1' , '01-jan-2023');
insert into ride_history values ( 2,'11' , 'stadium', '11', '02-jan-2023');
insert into ride_history values ( 3,'12' , 'cafe', '111' , '03-jan-2023');
insert into ride_history values ( 4,'13' , 'bookshop', '1111' , '04-jan-2023' );
insert into ride_history values ( 5,'14' , 'school', '11111' , '05-jan-2023' );

select * from ride;
insert into ride values ( 10,'1' , '01-jan-2023', '01-jan-2023' );
insert into ride values ( 11,'11' , '02-jan-2023', '02-jan-2023');
insert into ride values ( 12,'111' , '03-jan-2023', '03-jan-2023' );
insert into ride values ( 13,'1111' , '04-jan-2023', '04-jan-2023' );
insert into ride values ( 14,'11111' , '05-jan-2023', '05-jan-2023' );

select * from rider_documentation;
insert into rider_documentation values ( '444555' ,'12' , '224466', '1');
insert into rider_documentation values ( '555666' ,'13' , '335577', '11');
insert into rider_documentation values ( '666777' ,'14' , '446688', '111');
insert into rider_documentation values ( '777888','15' , '556677', '1111');
insert into rider_documentation values ( '888999' ,'16' , '778899', '11111');
```

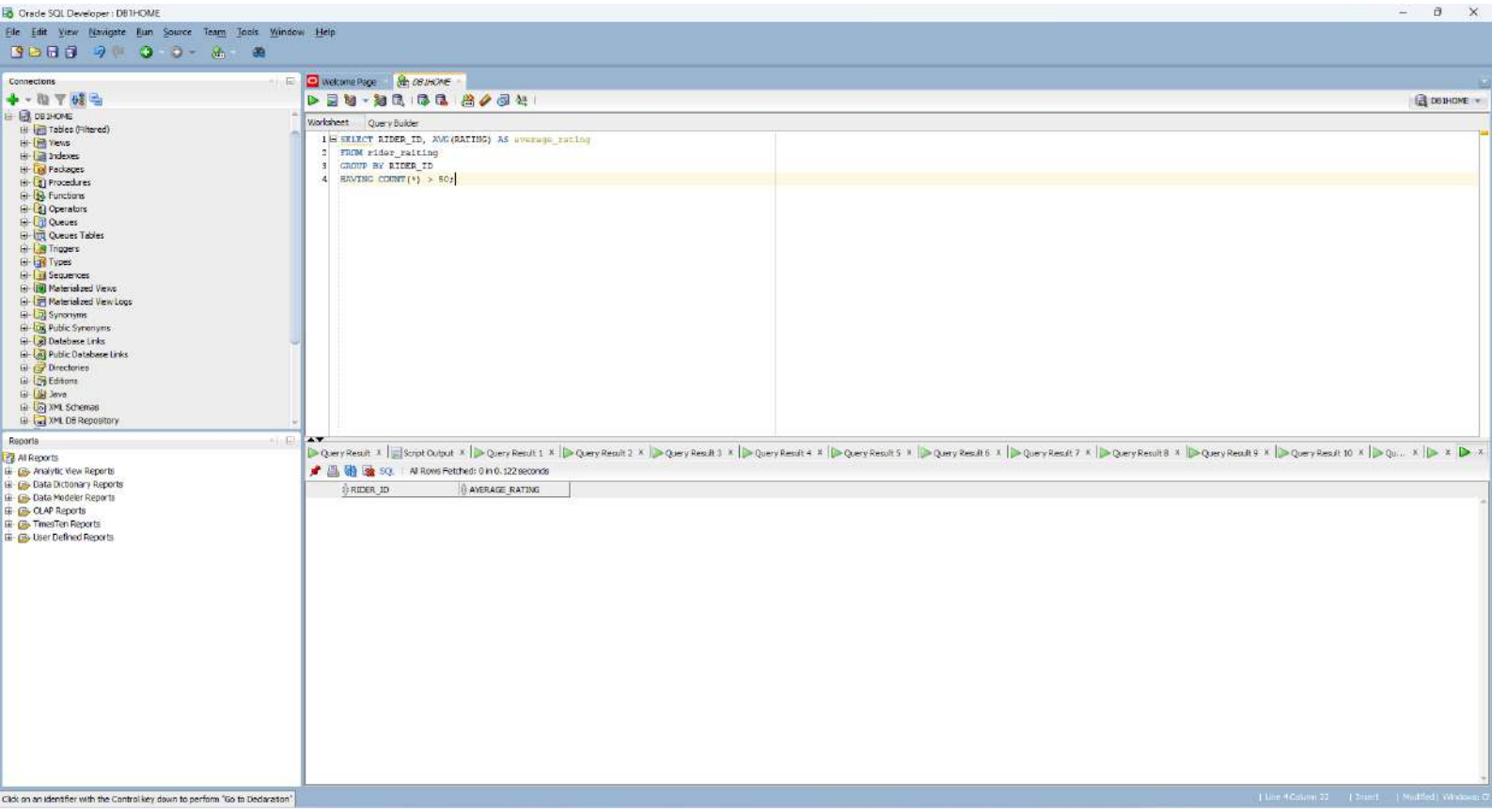
```
select * from rider_raiting;
insert into rider_raiting values ( '1' ,'10' , '1' , 'comment' , 'positive');
insert into rider_raiting values ( '2' ,'10' , '11' , 'comment' , 'positive');
insert into rider_raiting values ( '3' ,'7' , '111' , 'comment' , 'normal');
insert into rider_raiting values ( '4','9' ,  '1111' , 'comment' , 'positive');
insert into rider_raiting values ( '5' ,'8' , '11111' , 'comment' , 'normal');

insert into rider_schedule values ( '1' ,'01-jan-2023' , '01-jan-2023', '01-jan-2023' , '1');
insert into rider_schedule values ( '2' ,'02-jan-2023' , '02-jan-2023', '02-jan-2023' , '11');
insert into rider_schedule values ( '3' ,'03-jan-2023' , '03-jan-2023', '03-jan-2023' , '111');
insert into rider_schedule values ( '4','04-jan-2023' , '04-jan-2023', '04-jan-2023' , '1111');
insert into rider_schedule values ( '5' ,'05-jan-2023' , '05-jan-2023', '05-jan-2023' , '11111');

insert into salary values ( '1' ,'1' , '100', '50' , '150' , 'deposit');
insert into salary values ( '2' ,'11' , '150', '50' , '200', 'deposit');
insert into salary values ( '3' ,'111' , '120', '100' , '220', 'deposit');
insert into salary values ( '4','1111' , '110', '30' , '140', 'cash');
insert into salary values ( '5' ,'11111' , '170', '75' , '245', 'deposit');
```

SQL statements and Outputs

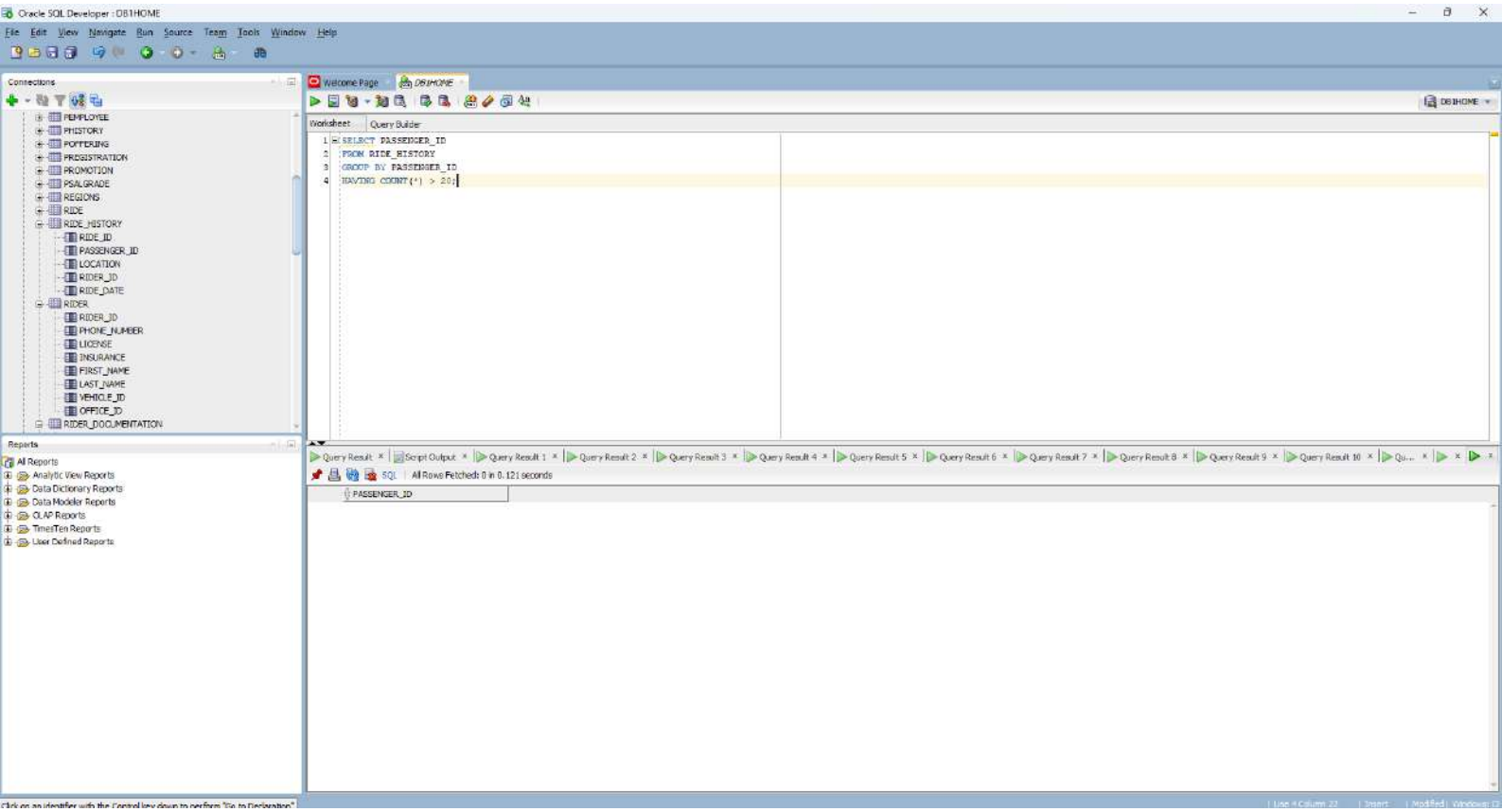
1: SELECT RIDER_ID, COUNT(ride_id) AS total_rides
FROM ride_history
GROUP BY RIDER_ID;



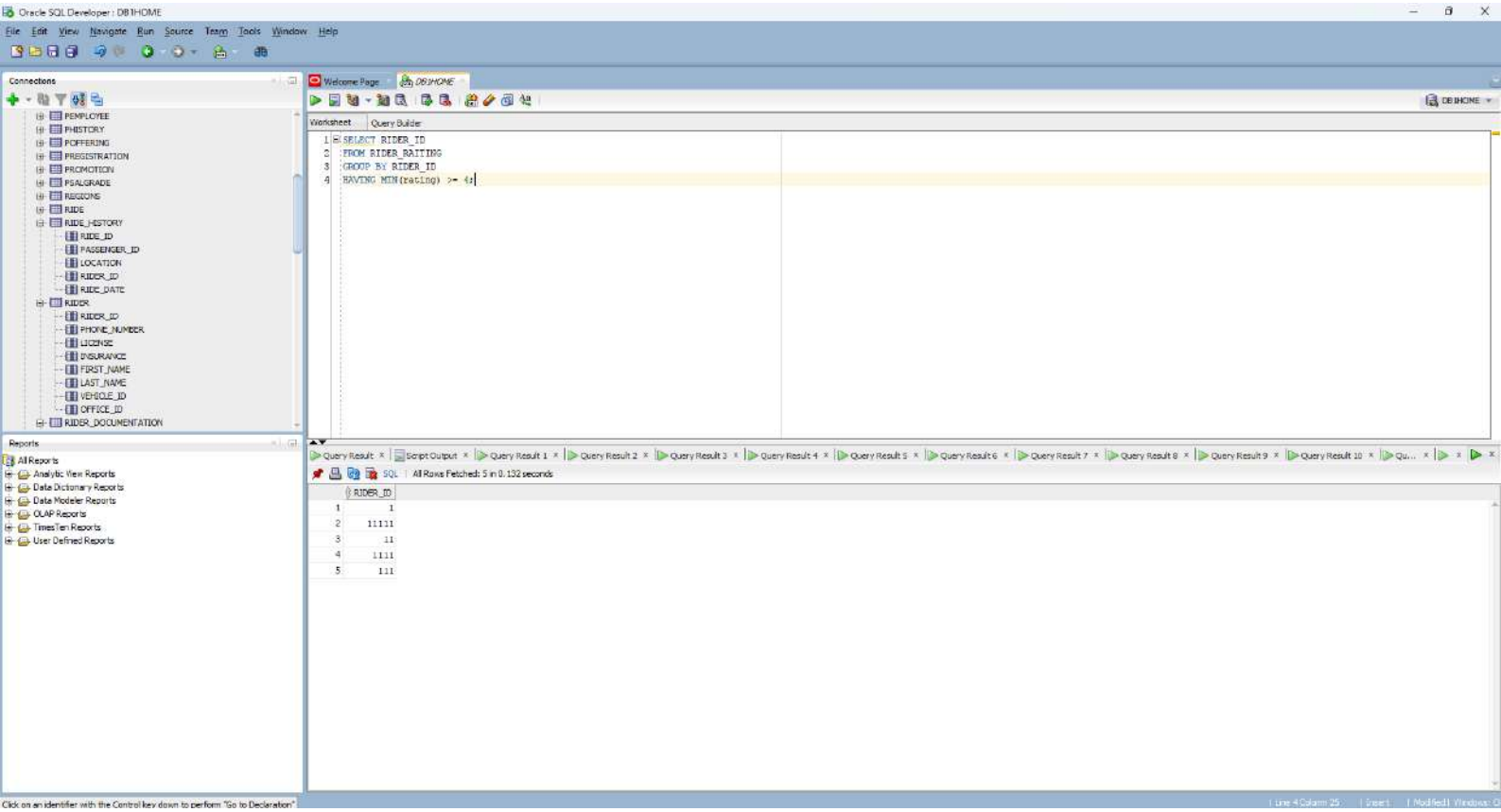
2: SELECT RIDER_ID, AVG(RATING) AS average_rating
FROM rider_raiting
GROUP BY RIDER_ID
HAVING COUNT(*) > 50;



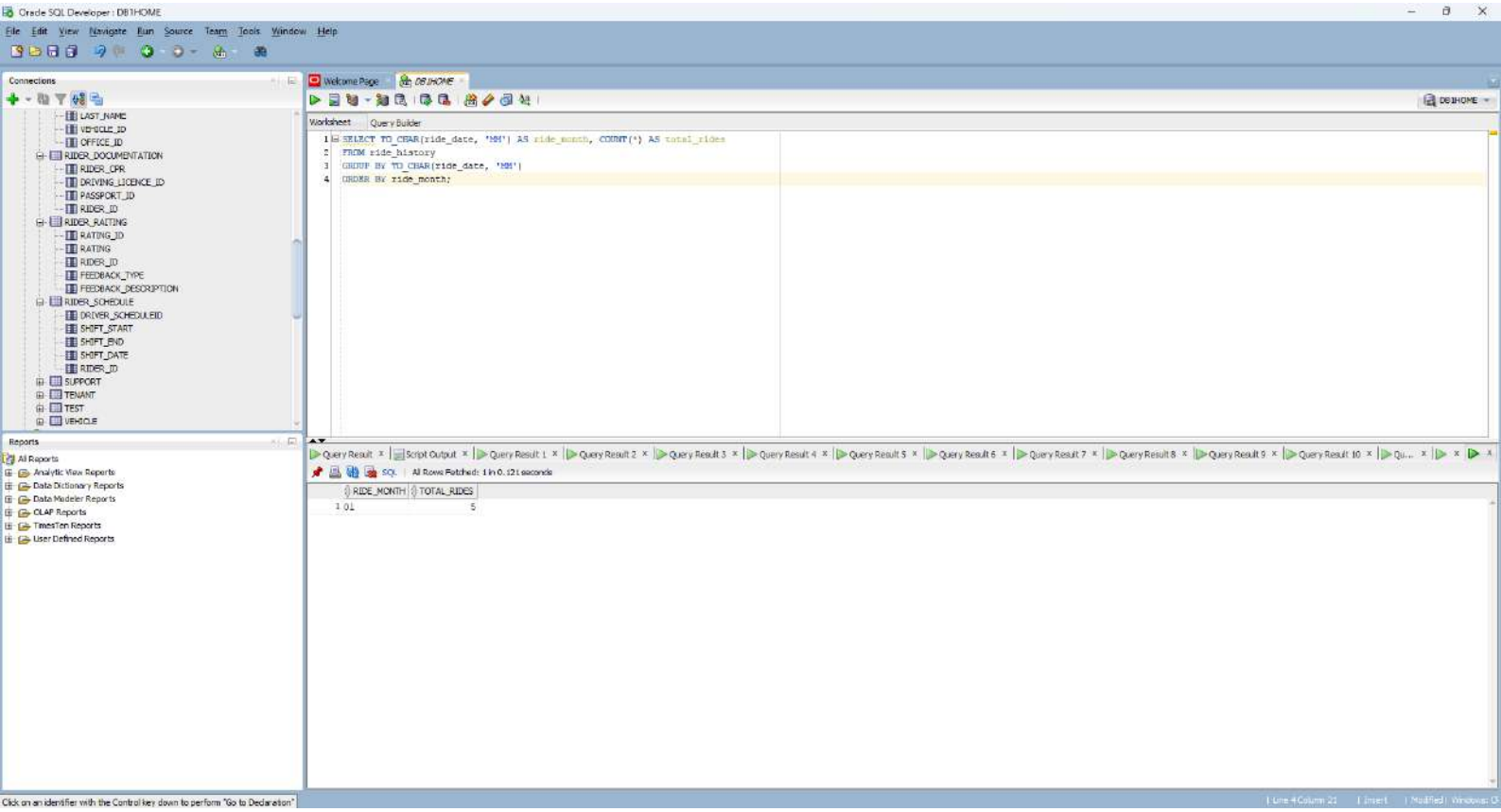
3: SELECT PASSENGER_ID
FROM RIDE_HISTORY
GROUP BY PASSENGER_ID
HAVING COUNT(*) > 20;



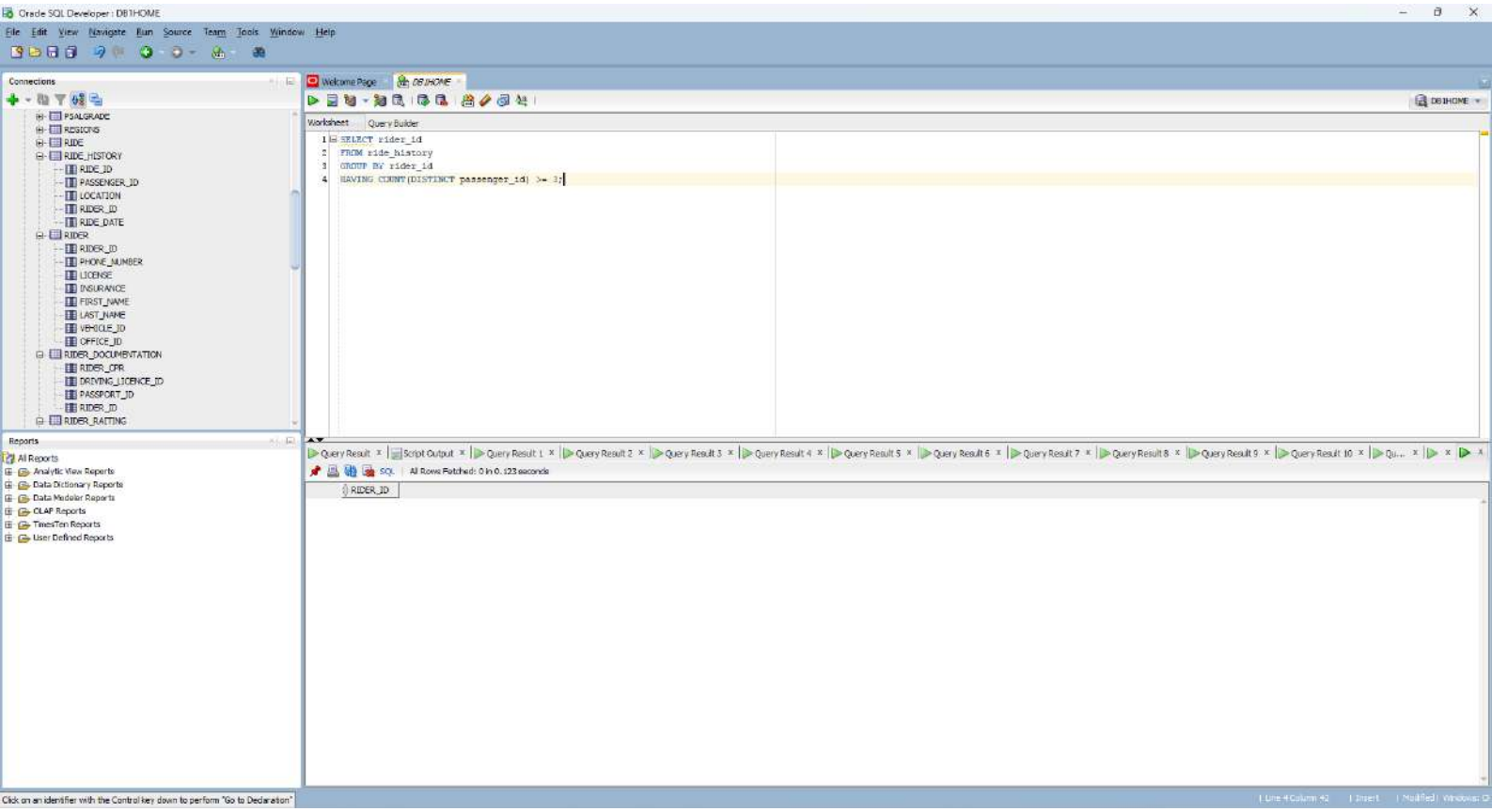
4: SELECT RIDER_ID
FROM RIDER_RAITING
GROUP BY RIDER_ID
HAVING MIN(rating) >= 4;



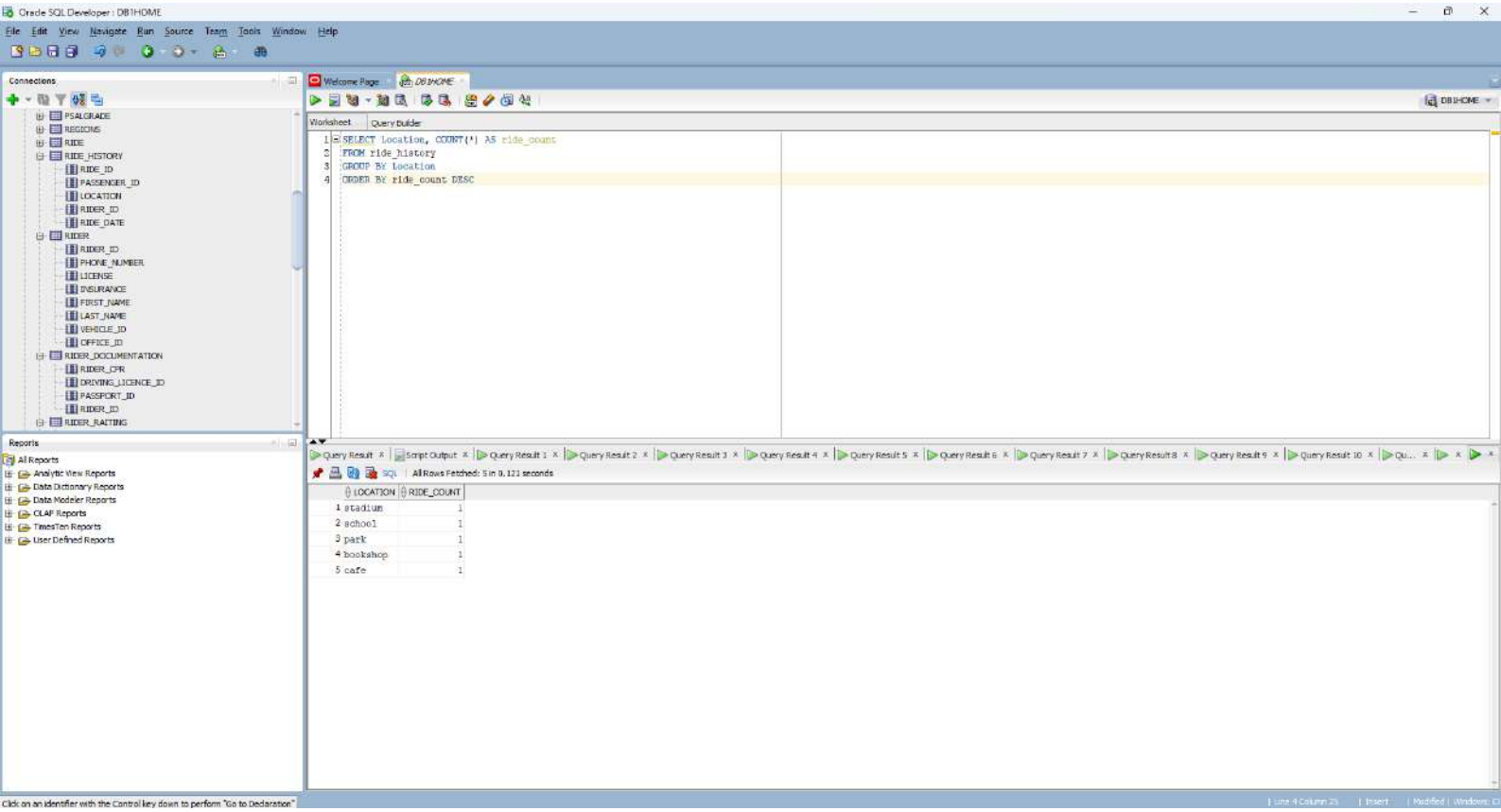
5: SELECT TO_CHAR(ride_date, 'MM') AS ride_month, COUNT(*) AS total_rides
FROM ride_history
GROUP BY TO_CHAR(ride_date, 'MM')
ORDER BY ride_month;



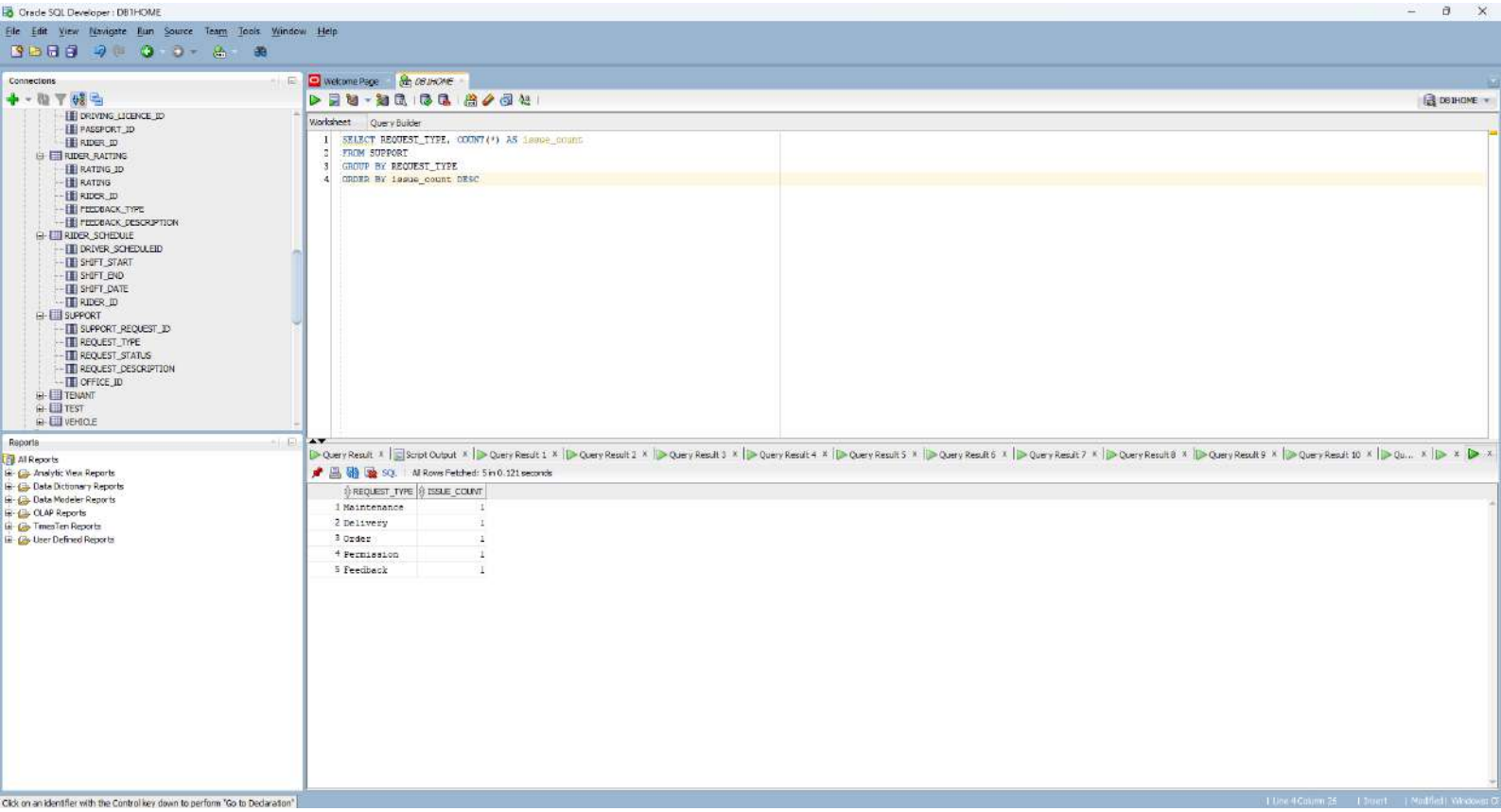
6: SELECT rider_id
FROM ride_history
GROUP BY rider_id
HAVING COUNT(DISTINCT passenger_id) >= 3;



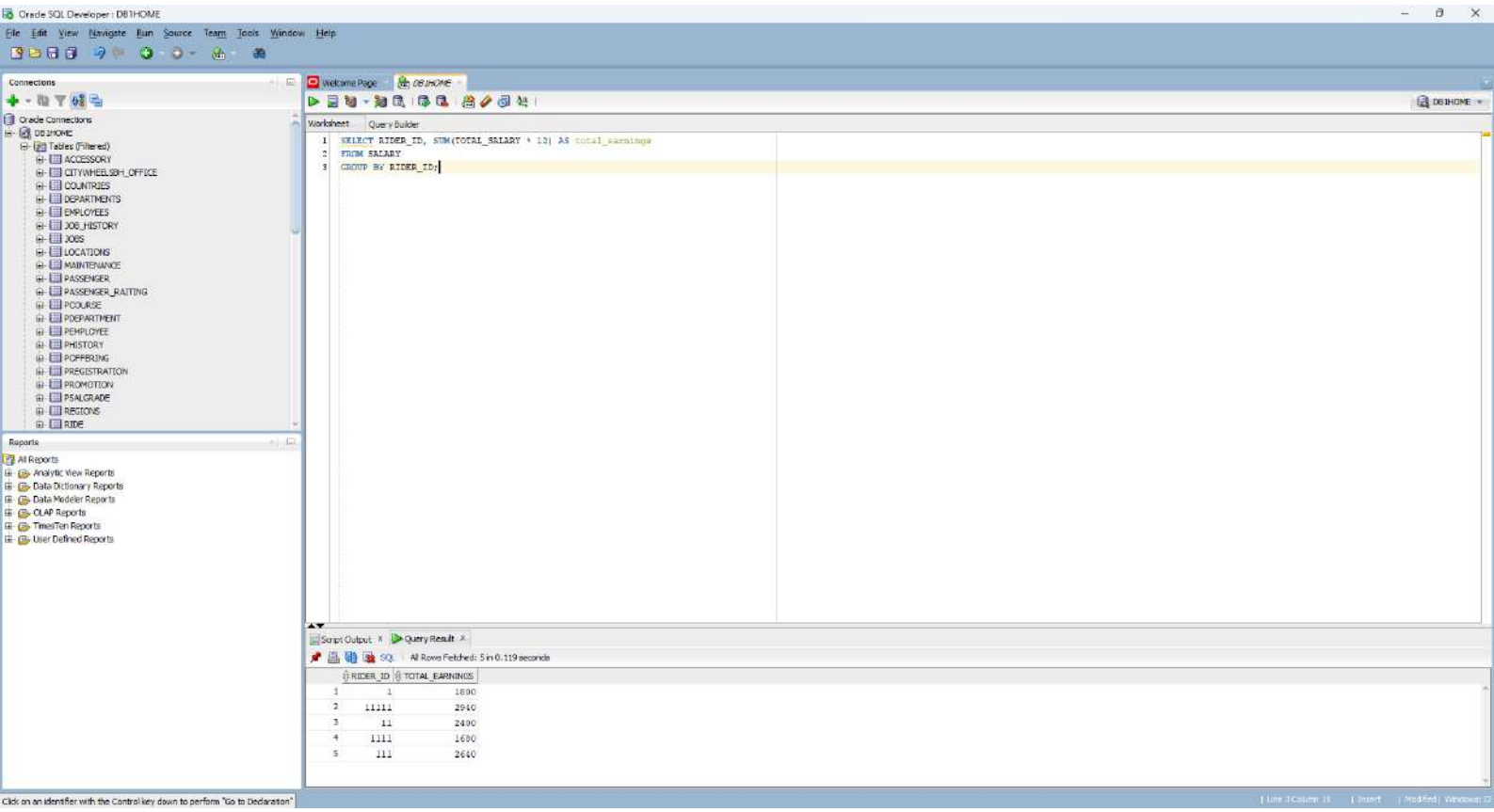
7: SELECT Location, COUNT(*) AS ride_count
FROM ride_history
GROUP BY Location
ORDER BY ride_count DESC



8: SELECT REQUEST_TYPE, COUNT(*) AS issue_count
FROM SUPPORT
GROUP BY REQUEST_TYPE
ORDER BY issue_count DESC



9: SELECT RIDER_ID, SUM(TOTAL_SALARY * 12) AS total_earnings
FROM SALARY
GROUP BY RIDER_ID;



10: SELECT * FROM PROMOTION
WHERE EXPIRATION_DATE >= SYSDATE AND EXPIRATION_DATE <= SYSDATE + 30;